空中加油问题的讨论

一． 问题重述

空中加油技术可以大大提高飞机的直航能力。作战飞机称为主机，加油机称为辅机。已知：（1）主机和辅机载油量、速度、单位时间的耗油量完全一样，且为常数；（2）飞机载油量可供飞行 L 公里；（3）辅机可以给主机或其他辅机加油；（4）执行完任务后，所有飞机必须返回基地；（5）飞机的起飞、降落、转向、加油的耗时和主机执行任务的时间忽略不计。

A 空军基地有一架主机和 n 架辅机，主机最大作战半径指主机在辅机加油协助下能飞到（并安全返回）离基地 A 的最远距离。有如下问题：

问题 1：每架飞机只能上天一次，求 n=1,2,3,4 时的最大作战半径r_n。

问题 2：n>4 时求r_n或给出r_n的上下界；讨论$n\to\infty$时r_n的渐进关系；给出判断最优作战方案（主机能飞到r_n处）的必要条件或充分条件。

问题 3：每架辅机可以多次上天，辅机从在机场降落到再次上天的时间间隔至少为相当于飞行 L/12 的时间，求最大作战半径R_n.

问题 4：另有两个待建的空军基地A_1，A_2，主机必须从基地 A 起飞，在基地 A 降落，辅机可以在任一基地待命和降落，可以多次起飞，在问题 3 的假设条件下讨论A_1，A_2的选址和主机最大作战半径R_n^*。

问题 5：ABCD 为矩形，AB=4L，AD=2L。A，B，D 为空军基地，主机从 A 起飞，到 C 执行任务，再返回 A。在问题 3 的假设条件下按最快到达并返回和最少辅机架数两种情况给出作战方案。

二． 模型假设

1、 辅机只有一个载油箱，容积为飞行距离 L 公里所需的油。载油箱的油可以用于自己飞行也可以给其他飞机加油。
2、 不考虑加油的安全因素，即对整个空域来说，辅机都是安全的。
3、 辅机可以同时给任意架飞机同时加油，加油耗时忽略不计。

三． 问题分析

空中加油问题是个在一系列约束下求最优的问题。目标是使作战半径最大，变元是作战方案，约束包括油料、地点和时间三个因素。制约主战半径的核心因素是油料，主机飞行所需油料除了起飞时自身载油外，只能靠辅机空中加油。空中加油最基本的约束是加油机和受油机必须出现在同一地点，这就使空中加油问题受时间、地点因素的制约。

在问题 1，2 的假设条件下，由于飞机只能上天一次，上天时间任意，故根据加油的地点即可判断出飞机的起飞时间，因此时间和地点因素可以统一为地点因素。在一些基本定理条件下，给定加油关系，可以根据油料约束计算加油地点，

进而计算作战半径。问题 3，4，5 中，辅机可以多次上天，在机场降落和再次上天的时间间隔因素不能和地点因素统一，必须另外考虑，因此问题变得十分复杂。我们在问题 1，2 结论的基础上给出一些松弛条件，使问题简化，搜索空间变成加油方案的子集，寻找接近最优的加油方案。

为了方便说明，我们先给出作战方案的二叉树模型。

我们将作战方案建模成如图 1 所示的二叉树结构。横坐标 t 轴代表飞行时间，纵坐标 x 轴代表飞离基地的距离。假设飞机的飞行速度为 v，将纵坐标的单位取为 L，横坐标的单位取为 L/v，则二叉树的所有线段斜率绝对值相同，且都为 1，则飞机的飞行路径在坐标系中表现为等腰三角形。二叉树的叶子结点纵坐标都为 0，代表飞离基地或返回基地。根结点代表主机的折回点。其他分叉结点是加油点。

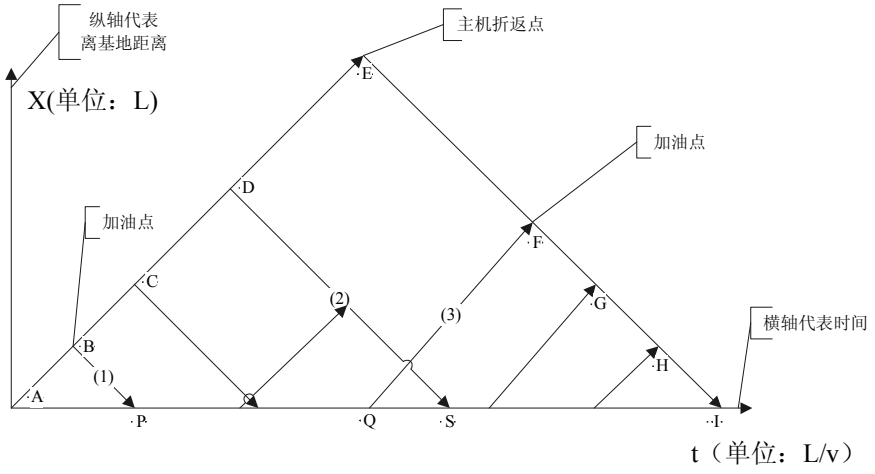


图 1　作战方案的二叉树模型

则图 1 作战方案中，主机飞行路径为 A→B→C→D→E→F→G→H→I，从 A 点起飞，到达离基地最远点 E 折回，到 I 点返回基地。作战半径为 E_x，总的飞行时间为 I_t，显然 $I_t = 2E_t$。同理，（1）号辅机的飞行路径为 A→B→P，在最远点 B 给其他飞机加油，（3）号辅机飞行路径为 Q→F→G→H→I。所以，在 A→B 段有一架主机和 3 架辅机同时飞行，B→C 段是一架主机和 2 架辅机，G→H 段是一架主机和 2 架辅机，依次类推。

下面具体分析各个约束。由于飞机速度和单位时间的耗油量是个常数，因此可以将时间和油量都用距离表示，指代该时间段内飞行的距离和这些油量能支持飞机飞行的的距离。

（1）油料约束

图 2 中一架辅机的飞行路径为 A→B→C→D→E。设辅机在 B 点加油前油料为 x_0，在 B 点受油 x_{in1}，在 C 点给其他飞机加油 x_{out}，到了 D 点受油 x_{in2}。则应满足如下限制：

$$\begin{cases} x_0 + x_{in1} \le L \\ x_0 + x_{in1} - (x_1 - x_2) - x_{out} - (x_1 - x_3) \ge 0 \\ x_0 + x_{in1} - (x_1 - x_2) - x_{out} - (x_1 - x_3) + x_{in2} \le L \end{cases} \quad (1)$$

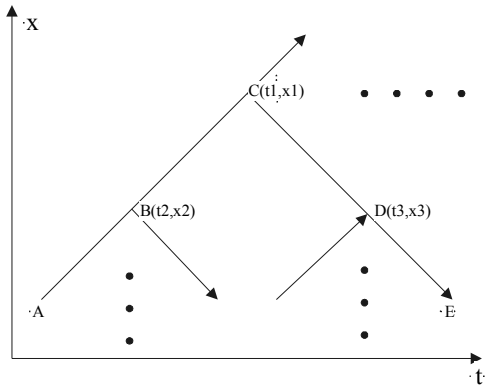


图 2　油料约束

（2）可统一的时间地点约束

主要指加油时加油机和受油机的时间地点应该一致，即加油机和受油机在坐标系中应有重合点，而且加油点在重合点中。

如图 1 中，主机和 1 号辅机的重合路径为 A→B，AB 上的点都是重合点，即 1 号辅机给主机的加油点只能在 A→B 区间内。

（3）辅机可以多次上天的时间约束

辅机降落和再次上天的时间间隔要大于等于 L/12，仍如图 1 中，（2）号辅机降落时间比（3）号辅机起飞时间晚，因此，（3）号辅机的任务不可能由（2）号辅机完成，而（1）号飞机降落时间早于（3）号飞机的起飞时间，当$|PQ|>\frac{1}{12}$时，3 号辅机的任务可以由 1 号辅机完成，（1）号辅机和（3）号辅机可以是一架飞机。

四．　模型准备

下面三个结论是问题 1，2 条件下最优作战方案的分析结论，也是解决问题 1，2 两种搜索方法的基础。

结论 1：辅机只能一次上天条件下，最优作战方案主辅机执行完任务后返回基地时油料要用尽。

由于制约作战半径的主要因素是油料，使作战半径最大也就是使主辅机携带的油料尽量充分利用于飞行。

结论 2：最优作战方案中辅机飞行距离尽可能短。

因为加油的目的是使主机飞得更远，辅机的飞行是无效飞行。辅机的飞行距离尽可能短，油料用于主机飞行的才最多，作战半径才可能尽量大。

结论 3：加油点可以等效为在二叉树的结点处。每个结点从左分枝到右分枝结点的油由一架飞机提供。

这个结论可由结论 1 和结论 2 推导出。

如图 2 中，假设 B→C 段有 K+1 架飞机飞行，K+1 架飞机在 B 点加油后油箱都是满的，到 C 点后辅机把另外 K 架飞机的油箱加满，飞到 D 点，辅机油料用尽，要靠其他辅机加油才能飞回基地。在这个结论下，油料关系（1）式的前两个不等式成了等式约束。

$$\begin{cases} x_0+x_{in1}=L \\ x_0+x_{in1}-(x_1-x_2)-x_{out}-(x_1-x_3)=0 \end{cases} \qquad (2)$$

因为 $x_{out}=K(x_1-x_2)$，所以（2）式即为

$$(K+1)(x_1-x_2)+(x_1-x_3)=L \qquad (3)$$

即每个结点的左分枝和右分枝的油由一架飞机提供。

但是，由公式（3）得：$x_1=\dfrac{L+x_3+(k+1)*x_2}{k+2}$;还需要满足实际情况：

$x_1\geq x_2, x_1\geq x_3$。

（3）式是计算结点坐标的基本公式。根结点的纵坐标值就是作战半径，其余各结点的纵坐标值是加油点的位置。问题 1，2 求解的两种方法都是在给出作战方案的二叉树结构的基础上利用该公式计算作战半径，比较作战方案的优劣的。

五． 模型建立与问题求解

作战方案采用二叉树模型，模型已经在问题分析部分做了说明。

二叉树模型中各结点的坐标按公式（3）计算。由于纵横坐标单位做了特别规定，坐标系中各点对应的纵坐标和横坐标数值相等，即只需计算纵坐标即可。

1、问题 1，2

可以采用二叉树穷举搜索方法和叶子结点生长方法两种方法搜索 n 架辅机时的最优作战方案，其对应的作战半径就是最大作战半径 r_n。

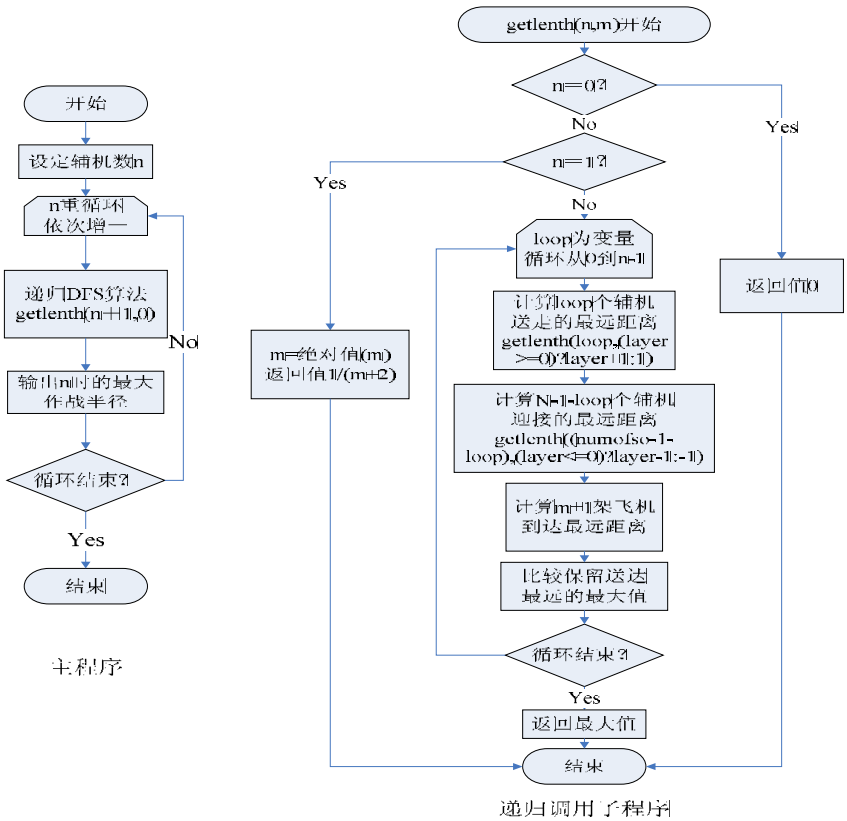


图 3　二叉树穷举搜索流程图

（1）二叉树穷举搜索方法

注意到二叉树模型中，飞机（包括主机和辅机）的总架数就是非叶子结点的数目。因此 n 架辅机时，搜索所有非叶子结点数目为 n+1 的二叉树的所有结构，按照公式（3）计算出各个结点的坐标，由此得到作战半径。使作战半径最大的二叉树结构就是最优作战方案，对应的作战半径就是所求的r_n。

程序根据二叉树的数据结构，采用回溯的方法实现。

流程图见图 3。主要程序见附录三。

程序计算的 n=1~22 的最大作战半径见表 1。

表 1 最大作战半径r_n

辅机架次 n	主机作战半径r_n	辅机架次 n	主机作战半径r_n
1	0.6666667	12	1.3333333
2	0.8333333	13	1.3611111
3	0.9166667	14	1.3888889
4	1	15	1.4126984
5	1.0555556	16	1.4365079
6	1.1111111	17	1.4550265
7	1.1611111	18	1.473545
8	1.2111111	19	1.4914021
9	1.2444444	20	1.5092593
10	1.2777778	21	1.5259259
11	1.3055556	22	1.5425926

对应的 n=1~4 时的最优作战方案见图 4，n＝10 时的最优作战方案见图 5。

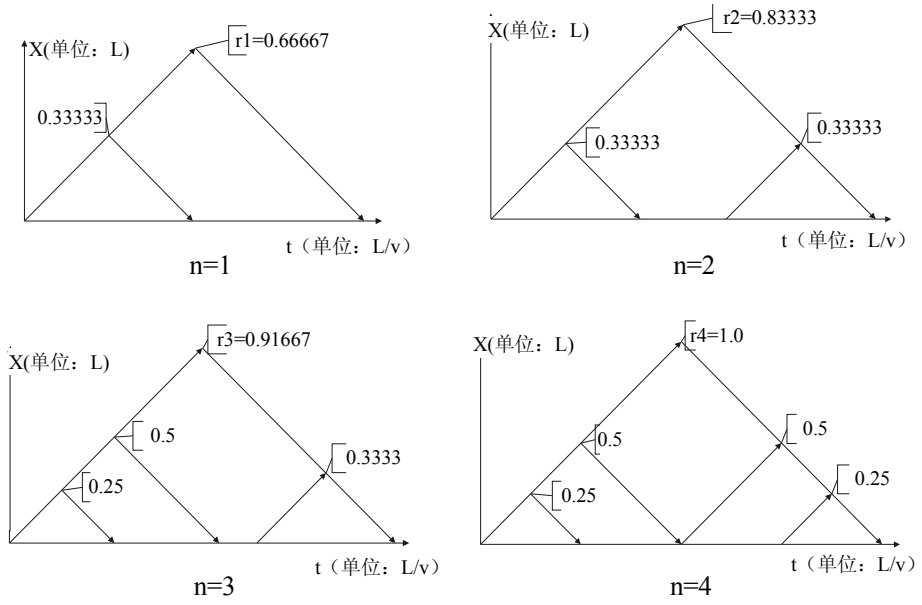
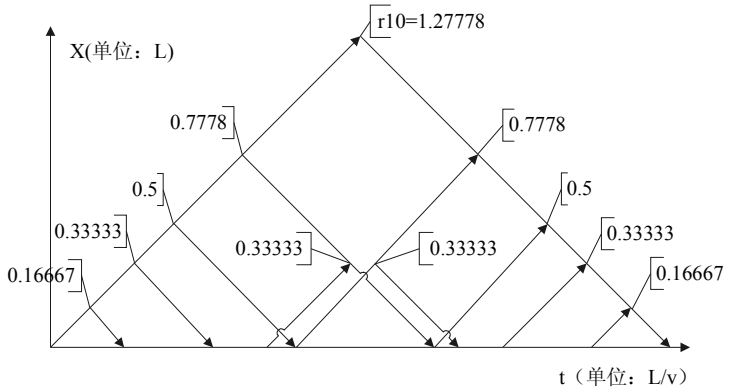


图 4　n=1~4 的最优作战方案



n=10

图 5　n=10 的最优作战方案

这种搜索方法的优点是容易理解，肯定能得到最大作战半径和最优作战方案。缺点是时间复杂度过大。当 n=23 时，搜索时间已经超过 1 个小时，n>23 时间长得不能容忍。为了改善这一点，我们探索除了时间复杂度小的叶子结点生长方法。

（2）叶子结点生长方法

该方法正确性的核心是定理 1 成立。

定理 1：（n+1）架辅机的最优的作战方案二叉树是 n 架辅机的最优作战方案二叉树的某个叶子结点生出两个叶子结点。

定理 1 简单情况时的证明见附录一。

这里给出说明，设 n=2 时的最优作战方案是图 6（1）所示的二叉树，则某个叶子生出两片叶子的情况有四种：图 6（2）~（5）。也即 n＝3 时的最优作战方案必然是这四种情况中的一种。计算可以发现图 6（2）和图 6（5）都是最优的。

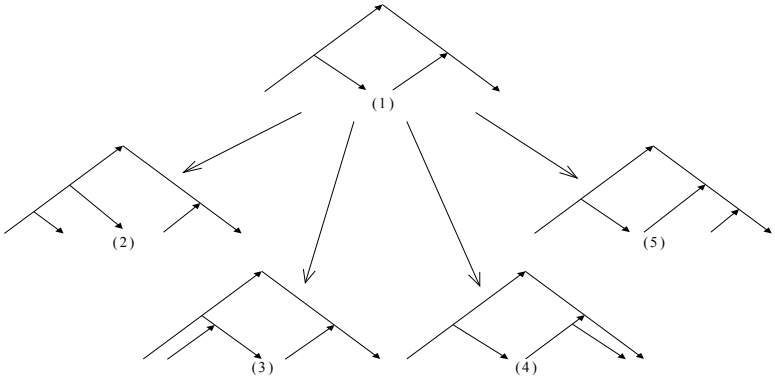


图 6　叶子结点生长情况

根据定理 1，搜索（n+1）架辅机的最优作战方案时，只要搜索 n 架辅机最优作战方案的某叶子结点生出两片叶子的所有可能情况就可。搜索空间比二叉树穷举搜索减小。时间复杂度降低。

程序的流程图如图 7。主要程序见附录四。

计算出的 n＝1~48 的最大作战半径见表 2

n=1~4 时的最优作战方案和图 4 完全相同。

比较表 1，表 2 还可以看出，得到的数据完全相同，这也进一步验证了叶子结点生长方法的正确性。

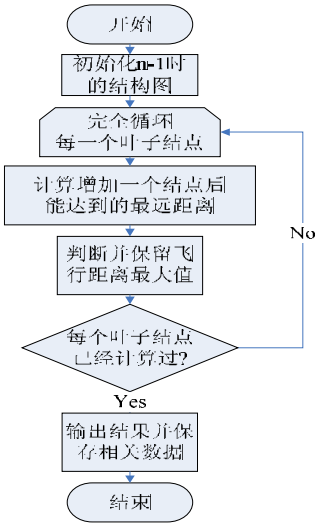


图 7 叶子结点生长方法流程图

表 2　最大作战半径 r_n

辅机架次 n	主机作战半径 r_n	辅机架次 n	主机作战半径 r_n	辅机架次 n	主机作战半径 r_n	辅机架次 n	主机作战半径 r_n
1	0.6666667	13	1.3611111	25	1.5898148	37	1.7259259
2	0.8333333	14	1.3888889	26	1.6037037	38	1.7351852
3	0.9166667	15	1.4126984	27	1.6175926	39	1.7444444
4	1	16	1.4365079	28	1.6314815	40	1.7537037
5	1.0555556	17	1.4550265	29	1.6425926	41	1.7627946
6	1.1111111	18	1.473545	30	1.6537037	42	1.7718855
7	1.1611111	19	1.4914021	31	1.6648148	43	1.7802189
8	1.2111111	20	1.5092593	32	1.6759259	44	1.7885522
9	1.2444444	21	1.5259259	33	1.687037	45	1.7968855
10	1.2777778	22	1.5425926	34	1.6981481	46	1.8052189
11	1.3055556	23	1.5592593	35	1.7074074	47	1.8131554
12	1.3333333	24	1.5759259	36	1.7166667	48	1.8210919

从上述两种方法的计算结果可以看出，n 增大时，r_n 也增大，但增大幅度越来越小。那么 r_n 有没有极限呢？

定理 2：当 $n\to\infty$，最大作战半径 $r_n\to\infty$。

该定理的构造性证明见附录二。

r_n 的一个下界是 $\frac{1}{2}+\frac{1}{3}\lfloor\log_3(n+1)\rfloor$，一个上界是 $\frac{1}{6}+\frac{1}{3}\lceil\log_2(n+2)\rceil$。即 $\frac{1}{2}+\frac{1}{3}\lfloor\log_3(n+1)\rfloor\le r_n\le\frac{1}{6}+\frac{1}{3}\lceil\log_2(n+2)\rceil$。其中 $\frac{1}{2}+\frac{1}{3}\lfloor\log_3(n+1)\rfloor$ 代表不大于 $\log_3(n+1)$ 的最大整数，$\lceil\log_2(n+2)\rceil$ 表示不小于 $\lceil\log_2(n+2)\rceil$ 的最小整数。

下界可以直接从定理 2 的构造性证明中得到。

上界的求出利用了定理 1。根据定理 1，n 增大时，二叉树通过扩展叶子结点使二叉树深度加大。我们知道，扩展一层叶子结点时，该叶子结点的纵坐标最大增加$\frac{1}{3}$。假设每扩展一层结点纵坐标都增加$\frac{1}{3}$，则最优的作战方案将是完全二叉树。除根节点外的每层深度为$\frac{1}{3}$的完全二叉树，n 架辅机可以排到$\lceil \log_2(n+2) \rceil$层，这显然优于所有方案，故$r_n \le \left(\frac{1}{2}-\frac{1}{3}\right)+\frac{1}{3}\lceil \log_2(n+2) \rceil$。得到了上界。

图 8 给出了最大作战半径，上下界之间的关系。

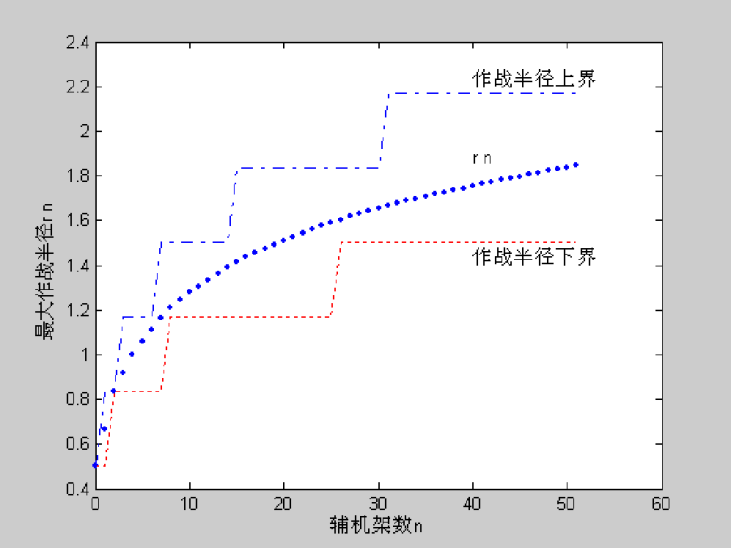


图 8　最大作战半径和上下界

下面给出三条最优作战方案的必要条件。

条件 1：结论 1，2，3。

条件 2：定理 1。

这个条件描述了 n 架辅机得最优作战方案和（n+1）架辅机最优作战方案之间得关系。

条件 3：二叉树根结点左子树和右子树完全对称（n 为偶数）或基本对称（n 为奇数）。

这可以从计算结果中得到验证。根节点代表主机，左子树是为主机前进服务的加油方案，右子树是为主机返回服务的加油方案。左子树的辅机是将满载油料的主机运的尽量远，右子树的辅机是将尽量远处的油料用尽的主机接回基地。这在本质上是一样的任务。故需要的辅机数量和加油方案应该基本相同。

2、问题 3

辅机可以多次上天的前提下，辅机多一次上天就可以多携带一次油料，这种情况下，辅机可能为了多一次上天，而使某几次加油任务完成返回时油料没有用完。即结论 1，结论 3 不再成立。这将使问题大大的复杂化。

我们先利用试凑的方法推出了 n=1~3 时的使作战半径最大的作战方案，考虑到方法的通用性，我们给出两个松弛条件，搜索满足松弛条件的接近最优的作战方案。

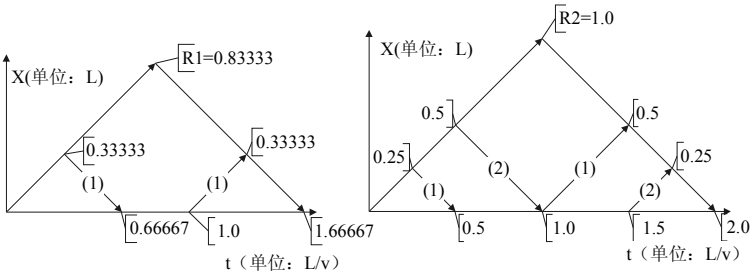
（1）试凑法

主要原则是在油料尽量充分利用的前提下让飞机多上天，得到了 n=1~3 时的最大作战半径和最优作战方案。

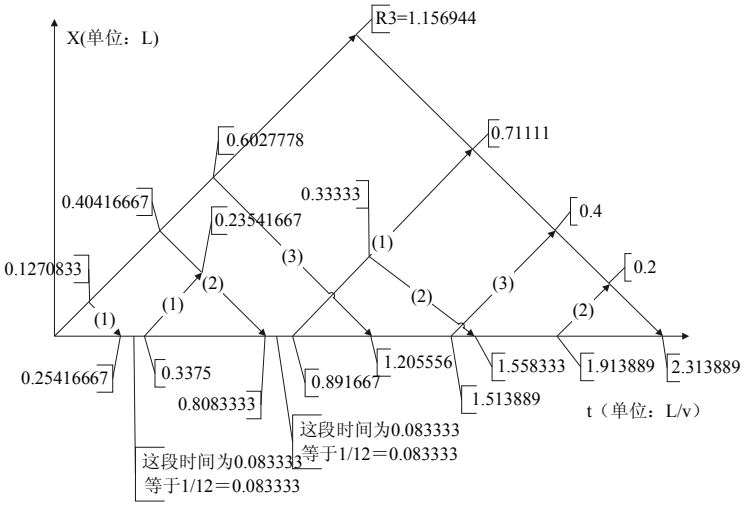
最大作战半径见表 3。对应的最优作战方案见图 9。

表 3　最大作战半径 R_n

辅机 n 架	作战半径 R_n
1	0.833333333333333
2	1.000000000000000
3	1.156944444444444



n=1　　n=2



n=3

图 9　n=1~3 的最优作战方案

这种方法由于没有固定规律，对 n=4 时已很难发挥作用。我们希望给出一种结果次优但比较实用的搜索方法。

（2）次优搜索法

为了使问题简化，借鉴解决问题 1，2 的结论提出如下两个松弛条件。

松弛条件 1：结论 1，3 仍然成立。

松弛条件 2：（n+1）架辅机的最优作战方案二叉树是 n 架辅机最优作战方案二叉树叶子结点的一层生长。

这个条件和定理 1 并不一样。

如图 10，n＝0 时，叶子结点的一层生长有三种情况，而定理 1 某个叶子结

点生出两个叶子结点的情况只是其中的（1），（3）。

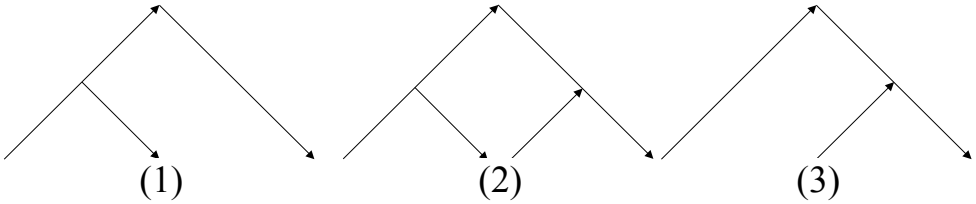


图 10　叶子结点生出一层的情况

在这两个松弛条件下，可以在解决问题 1，2 的叶子结点生长方法的基础上加进时间计算和能否重新上天的判断，并扩展了搜索空间。程序流程图见图 11。

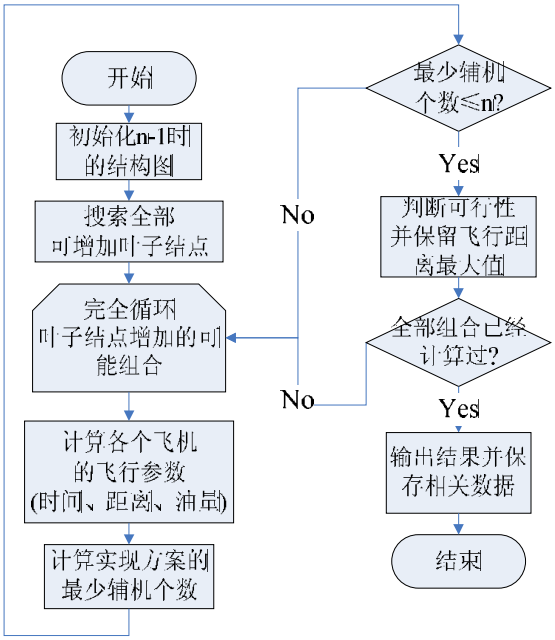


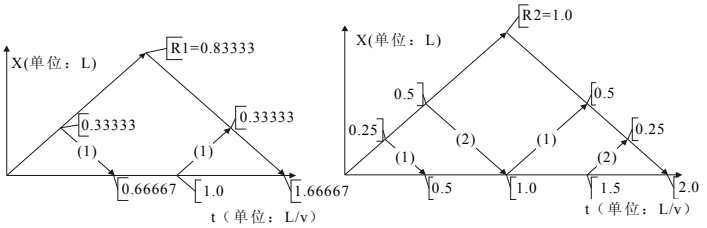
图 11　次优搜索方法流程图

计算的 n=1~9 的次优作战半径 $\hat{R}_n$ 如表 4。n=1~4 的次优作战方案见图 12。

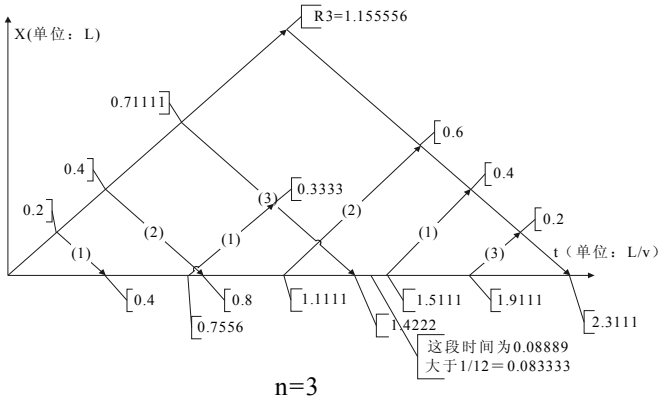
显然这种情况下算出的 n＝3 时的作战方案没有试凑法优。该方法的时间复杂度仍然比较大，当 n>9 时，时间之长不能容忍。

表 4 次优作战半径 $\hat{R}_n$

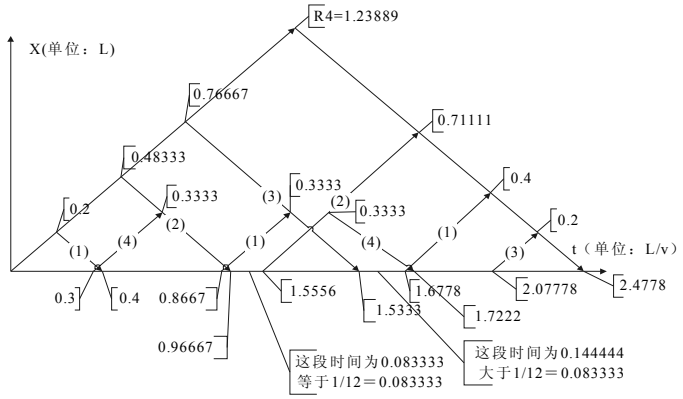
辅机 n 架	作战半径 $\hat{R}_n$
1	0.833333333333333
2	1.000000000000000
3	1.155555555555556
4	1.238888888888889
5	1.266666666666667
6	1.359259259259259
7	1.421957671957620
8	1.464285714285714
9	1.520370370370370



n=1　　n=2



n=3



n=4

图 12　n=1~4 的次优作战方案

3、问题 4

这一问是讨论 A1,A2 的选址方案使得主机作战半径最大。问题等同于求出在有基地能够支持主机续航的条件下的最大作战半径。

首先可以知道，最优作战方案中，基地 A ，A1 和 A2 应该在一条直线上，且它们之间的距离至少为 L。为了分析问题的方便，我们不考虑辅机从一个基地起飞，在另外一个基地降落的情形，则问题转化为如下两个方面：（1）n 架辅机在基地 A，A1，A2 上的分配方案 n_0，n_1 和 n_2；（2）在 n_0，n_1 和 n_2 已知的情况下，基地 A，A1，A2 上辅机的最优加油方案。

为了考虑问题的方便，我们不考虑主机可以在基地上空附近盘旋等油的情形，即主机一旦起飞，将沿着一条直线一直飞下去，直到返航。

首先解决 n_0，n_1 和 n_2 已知的情况下，基地 A，A1，A2 上辅机的最优加油方

案。在最优作战方案中，我们将基地 A 的辅机最远的加油点离基地 A 的距离定义为基地 A 的续航能力 R_A^*，基地 A1 或 A2 上的辅机两个相距最远的加油点之间的距离定义为基地 A1 或 A2 的续航能力 R_{A1}^* 或 R_{A2}^*。按照这种定义，$n_0=0$ 时，$R_A^*=0$；$n_1=1$ 时，$R_{A1}^*=0$；$n_2=1$ 时 $R_{A2}^*=0$。

为了搜索的方便，我们仍然给出条件使问题松弛。

松弛条件 3：即为问题 3 中次优搜索方法的松弛条件 1，2。

松弛条件 4：三个基地上的辅机加油方案中为主机前进服务的加油方案和为主机返回服务的加油方案完全相同，结构对称。

在这两个条件下，主机最大作战半径 R_n^* 和我们定义的续航能力之间的关系是 $R_n^*=R_A^*+1+R_{A1}^*+1+R_{A2}^*+0.5$。 **（4）**

首先我们编程搜索（程序略）R_A^*,R_{A1}^*,R_{A2}^* 和对应的 n_0,n_1,n_2 之间的关系，得到表 5。（由于程序时间复杂度大，只计算出如下几个数）

表 5　续航能力和辅机架数之间的关系

n_0	1	2	3	4	5
R_A^*	0.333333	0.5	0.766667	0.888889	0.922222
n_1	1	2	3		
R_{A1}^*	0	0.444444	0.888889		
n_2	1	2	3		
R_{A2}^*	0	0.444444	0.888889		

然后考虑 n 架辅机在基地 A，A1，A2 上的分配方案 n_0，n_1 和 n_2。显然对于每一种分配方案，可以根据表 5 查出三个基地的续航能力，然后根据公式（4）计算作战半径 R_n^*。因此对于 n 架辅机的所有分配方案，作战半径最大的分配方案就是最优分配方案。表 6 是 n=1~5 时的最大半径，辅机分配情况和基地选址情况。图 13 给出了 n＝5 时的作战方案。

表 6　n=1~5 的作战半径 R_n^*，辅机分配和基地选址

辅机架数 n	最大作战半径 R_n^*	基地 A 的辅机数 n_0	基地 A1 的辅机数 n_1	基地 A2 的辅机数 n_2	A1 到 A 的距离	A2 到 A 的距离
1	1.5	0	1	0	1	不建 A2
2	2.5	0	1	1	1	2
3	2.944444	0	1	2	1	0.444444
4	3.388889	0	2	2	1	2.444444
5	3.722222	1	1	3	1.333333	2.777778

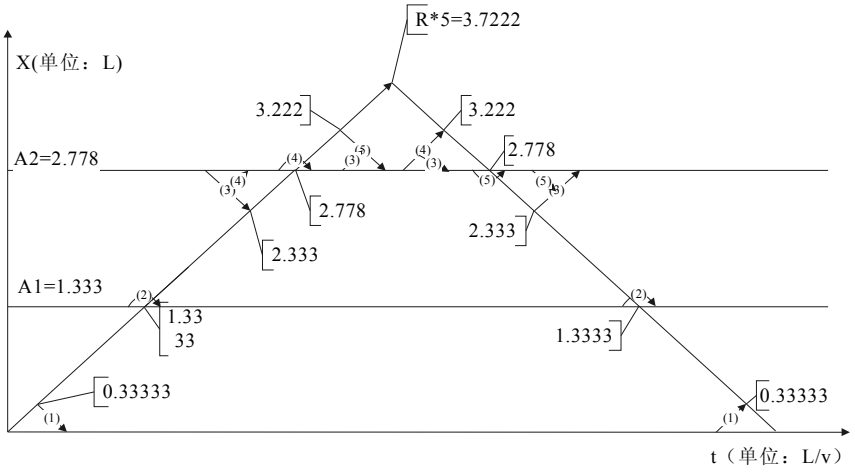


图 13 n＝5 时的作战方案

4、问题 5

显然，最快到达并返回的作战方案是使主机沿着对角线 AC 往返飞行；最优的加油方案是由基地 A 和基地 B 上的辅机加油。

六．　模型的进一步讨论

1、加油机的载油量问题

题目中假设加油机和主机的载油量相同，这一般是不符合实际的。如伊尔－78 空中加油机能够携带 118 吨燃料，KC－10A“致远”空中加油机具有 7 个油箱，最大供油量 90 吨。这都大于一般轰炸机和侦察机的载油量，更是一般战斗机载油量的几倍。

我们可以假设主机的载油量仍为 L，而辅机的载油量为 kL，k>1。在不考虑辅机安全因素的条件下，即辅机可以始终伴随主机飞行，下面给出辅机数量 n＝1～4 时，辅机只能升空一次的条件下的作战半径 r_n。

$$r_1=\begin{cases}\dfrac{k+1}{4}L, k\geq 3\\ \dfrac{k+3}{6}L, k<3\end{cases},\ r_2=\begin{cases}\dfrac{3k+2}{8}L, k\geq 6\\ \dfrac{2k+3}{6}L, k<6\end{cases},\ r_3=\begin{cases}\dfrac{2k+1}{4}L, k\geq 3\\ \dfrac{5k+6}{12}L, k<3\end{cases},\ r_4=\begin{cases}\dfrac{23k+10}{40}L, k>3\\ \dfrac{k+1}{2}L, k\leq 3\end{cases}$$

2、m 架主机，n 架辅机的情况

一般执行作战任务的主机都不是一架，因此更一般的情况是 m 架主机，n 架辅机的情况。

分析可以知道，这时最优的作战方案不是将 n 架辅机平均分成 m 组，每组分别辅助一架主机，而是将 m 架主机当成一个整体，利用 n 架辅机，一架主机时的作战方案加油，只是加油点和主机的作战半径都要发生变化。

下面给出飞机只能一次升空条件下，n=1~4 时的作战半径。

$$r_1^m=\frac{m+3}{2m+4}L,\quad r_2^m=\frac{m+4}{2m+4}L,\quad r_3^m=\frac{m^2+8m+13}{2m^2+10m+12}L,\quad r_4^m=\frac{m+7}{2m+6}L$$

3、经济因素

根据定理 2，当$n\to\infty$，最大作战半径$r_n\to\infty$，但对于远距离的作战任务采用大量的辅机加油是极不经济的。下面给出油料利用效率的概念，用于说明这个问题。

油料利用效率：主机飞行的总耗油量与作战方案中基地提供的油料总量之比。

显然在飞机只能上天一次的条件下，n 架辅机的最优作战方案的油料的利用效率为$\rho_n=\frac{2r_n}{(n+1)L}$。我们计算 n=0~9 时的油料的利用效率如下表。

辅机数量 n	0	1	2	3	4
油料效率 ρ_n	100%	33.3%	27.8%	22.9%	20%
辅机数量 n	5	6	7	8	9
油料效率 ρ_n	17.6%	15.9%	14.5%	13.5%	12.5%

可见，n 越大，油料利用效率ρ_n越低。因此若认为$\rho_n<20\%$是经济上不能容忍的话，辅机的数量就不应多于 4 架。

从问题 4 可以知道，数量少的辅机利用多个基地可以使主机的作战半径大大增加，油的利用效率也大大提高。因此当完成远距离的作战任务时，从经济角度要考虑建立空军基地。

七． 模型评价

优点：

（1） 论文将作战方案建模成二叉树结构，简单易懂，且方便编程；

（2） 对问题 1，2，3，论文都给出了两种寻找最优作战方案的方法，并比较了它们的优缺点；

（3） 给出了大量的计算结果和实现方案，数据真实可信；

（4） 解决问题 1，2 时由于穷举搜索的时间复杂度大，经过分析，提出了基于二叉树叶子结点生长的定理 1，并做了部分证明，进而给出了一种时间复杂度小的搜索方法，计算出 n=1~30 时的作战半

径。

（5） 给出了 $n \to \infty$， $r_n \to \infty$ 的定理，并给出了构造性证明；

（6） 结合实际，在进一步讨论中简单分析了加油机载油量大，多架主机和经济因素等问题，得到了一些结论。

缺点：

（1） 没有分析飞机盘旋的情况；

（2） 没有给出一种求解问题 3 最优作战方案的方法；

（3） 由于时间原因，对问题 5 没有深入讨论。

参考文献

[1] 姜启源编．数学模型(第二版)．北京：高等教育出版社．1996

[2] 吴翊，吴孟达，成礼智编著．数学建模的理论与实践．长沙：国防科技大学出版社．1999

[3] 施光燕，董加礼．最优化方法．北京：高等教育出版．2002

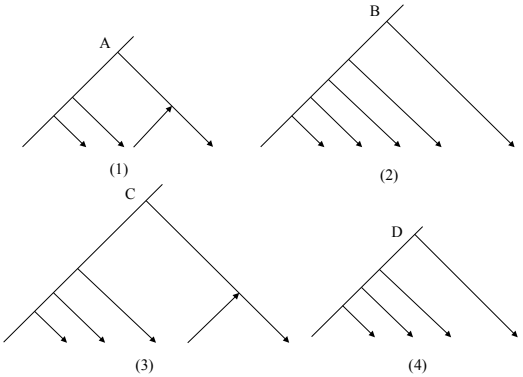
[4] 徐士良，朱明方编著．软件应用技术基础．北京：清华大学出版．2001

[5] 严蔚敏，吴伟民编著．数据结构(C 语言版)．北京：清华大学出版．2004

附录一　定理 1 简单情况的证明

定理 1：（n+1）架辅机的最优的作战方案二叉树是 n 架辅机的最优作战方案二叉树的某个叶子结点生出两个叶子结点。

反证法。如果（n+1）架辅机的最优作战方案不是 n 架辅机最优作战方案的叶子结点生出两片叶子，则可以证明另有 n 架辅机的作战方案优于最优作战方案，得出矛盾。



附录图 1

如附录图 1 中，（1），（2），（3），（4）是作战方案的接近叶子结点的某个小区域的分枝情况，其他部分完全相同。在 A,B,C,D 点返回的辅机都要为 K 架飞机加油。设（1），（4）是 n 架辅机的作战方案，（2），（3）是(n+1)架辅机的作战方案，（1），（2）是最优作战方案。显然，（3）是（1）的某个叶子结点生长得到的二叉树，（2）不是，即这种情况下不符合定理 1。下面推导矛盾。

可以计算出 A,B,C,D 四点的纵坐标是

$$A_x=\frac{10K+22}{3K^2+18K+24},\quad B_x=\frac{5}{K+6},\quad C_x=\frac{13K+29}{3K^2+21K+30},\quad D_x=\frac{4}{K+5}$$

由于其他部分相同，根据最优作战方案的条件，$A_x\ge D_x$，$B_x\ge C_x$。但这会得到$K\ge 4$和$K\le\sqrt{7}$的矛盾结论。

故在（1）是最优作战方案的前提下，不符合定理 1 的（2）不可能是最优作战方案。

当然，上述情况是一种简单情况，更复杂的情况课题同样证明。

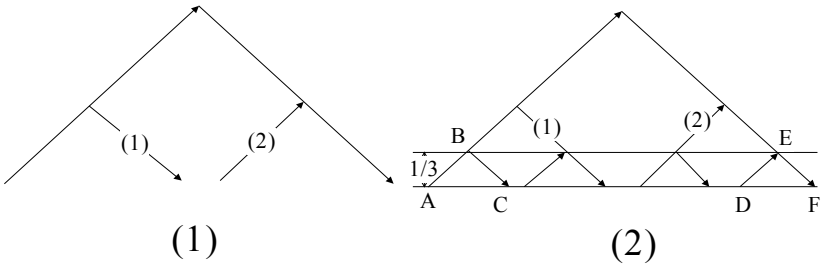
附录二 定理 2 的构造性证明

定理 2：当$n\to\infty$，最大作战半径$r_n\to\infty$。

假设（n-1）架辅机时最大作战半径为r_{n-1}。当有（3n-1）架辅机时采用如下方案可以使作战半径达到$r_{n-1}+\frac{1}{3}$。

给（n－1）架辅机和一架主机每架分配两架辅机负责接送，一架辅机负责将

满载油料的主机或辅机送到$\frac{1}{3}$远处，令一架负责将油料用尽的辅机或主机从$\frac{1}{3}$远处迎回基地。那（n－1）架辅机和一架主机采用最优作战方案飞行和加油，这样主机的飞行半径可以达到$r_{n-1}+\frac{1}{3}$。



(1)　　(2)

附录图 2

如附录图 2，n＝2 时的最优作战方案是（1），n＝8 时，采用上述方案飞行图见（2）。路径 A→B→C 和 D→E→F 上都有两架辅机，分别负责送主机和辅机 1 和接主机和辅机 2。

所以可以断定$r_{3n-1}\geq r_{n-1}+\frac{1}{3}$。

因为$r_0=\frac{1}{2}$，所以$r_{3^n-1}\geq\frac{1}{2}+\frac{n}{3}$。

考虑所有的 n 值，有$r_n\geq\frac{1}{2}+\frac{1}{3}\lfloor\log_3(n+1)\rfloor$，其中$\frac{1}{2}+\frac{1}{3}\lfloor\log_3(n+1)\rfloor$表示不超过$\log_3(n+1)$的整数。

附录三　二叉树穷举搜索主程序

```
#include "stdafx.h"
#include "solve.h"
double getlenth(int n,int m);
#define nbig 23                                          //定义最大循环次数
int main(int argc, char* argv[])
{
    for(int fu=0;fu<=nbig;fu++)
    {
        printf("(%16.14f)--(%d)\n",getlenth(fu+1,0),fu);    //计算并输出结果
    }
    return 0;
}

double getlenth(int n,int m)
{
    if(n==1)return 1./double(((m>=0)?m:-m)+2);          //辅机为 1 时返回 1/(abs(m)+2)
    else if(n==0)return 0.;                              //辅机为 0 时，返回 0
```

```
    else
    {
        double left,right,temp,biglenth=0;
        for(int loop=0;loop<n;loop++)                    //遍历全部可能，取出最大值
        {
            left=getlenth(loop,(m>=0)?m+1:1);            //递归计算 loop 架辅机送最远
            right=getlenth((n-1-loop),(m<=0)?m-1:-1); //递归计算 n-1-loop 迎接最远
            if(m>0)                                      //判断送还是接，计算最远距离
            {
                temp=left+(1+right-left)/(m+2);
            }
            else
            {
                int plane;
                plane=-m;
                temp=left+(1+(plane+1)*(right-left))/(plane+2);
            }
            if(temp>biglenth&&temp>=left&&temp>=right)biglenth=temp;
                                                 //验证 m 架飞机能够达到最远距离
                                                 //要大于 loop 架辅机送最远，同时
                                                 //大于 n-1-loop 迎接最远。
                                                 //比较、保留最大值。
        }
        return biglenth;                                 //返回最大值。
    }
}
```

附录四　叶子结点生长方法主程序

```
main()
{
    double a[mm][nn],b[mm][nn],aa;
    int row,col,i,j,k,n,xrow,xcol,br;
    for(i=0;i<mm;i++)//初始化二叉树结构, ,(0,0)点代表主机
        for(j=0;j<nn;j++)b[i][j]=0;
    b[0][0]=0.5;
    for(n=1;n<numoil;n++)//辅机数量从小到大循环求解,n 为辅机数量
    {
        aa=0;//设定很小的初始，用于记录最优解
        for(row=1;row<mm;row++)//循环添加 1 架辅机,row 为二叉树的层位置
            for(col=0;col<numcol(row);col++)//col 为二叉树的列位置
            {
                br=0;//用于检验是否符合实际
                for(i=0;i<mm;i++)//初始化为 n-1 架辅机
```

```
                    for(j=0;j<nn;j++)a[i][j]=b[i][j];
                if(a[row][col]==0 && a[row-1][col/2]>0)
                {//如果此节点无辅机且父结点有辅机，即可以添加此节点
              a[row][col]=1.0/(bbb(col*2,row+1)*1.0+bbb(col*2+1,row+1)*1.0);
                    j=col;//计算所有父结点的值
                    for(i=row;i>0;i--)
                  {
                        j=j/2;
a[i-1][j]=(1.0+a[i][j*2]*bbb(j*2,i)+a[i][j*2+1]*bbb(j*2+1,i))/(bbb(j*2,i)*1.0+bbb(j*2
+1,i)*1.0);
if(a[i-1][j]<a[i][j*2]-1e-12||a[i-1][j]<a[i][j*2+1]-1e-12)br=100;//不符合实际
                  }
                  if(aa<a[0][0]&&br==0)//记录最优解
                  {
                      aa=a[0][0];   xrow=row;   xcol=col;
                  }
                }
            }
          }
          //循环结束，计算最优解
           b[xrow][xcol]=1.0/(bbb(xcol*2,xrow+1)*1.0+bbb(xcol*2+1,xrow+1)*1.0);
         j=xcol;
         for(i=xrow;i>0;i--)
         {
             j=j/2;

b[i-1][j]=(1.0+b[i][j*2]*bbb(j*2,i)+b[i][j*2+1]*bbb(j*2+1,i))/(bbb(j*2,i)*1.0+bbb(j*
2+1,i)*1.0);
         }
     }
}//其中,numcol(row)为第 row 层结点的数量,bb(i,j)为 i 层第 j 个节点一同起飞或降
   //落的架数
```